

AMENDMENTS

In the Claims:

This listing of claims replaces all prior versions and listings of claims in the application.

1 1. (Currently Amended) A method for identifying infected program
2 instructions, comprising the steps of:
3 inserting a dynamic execution layer interface (DELI) between computing
4 device hardware and the program instructions;
5 monitoring the program instructions as they enter the DELI to determine if the
6 ~~code has program instructions have~~ been previously ~~processed by the computing~~
7 ~~device hardware~~ cached within the DELI, wherein the determination of whether the
8 program instructions have been cached is responsive to an association between native
9 application code and one or more analogues that have been transformed within the
10 DELI; and when it is the case that the ~~application code has program instructions have~~
11 not been previously ~~processed~~ cached within the DELI,
12 analyzing the program instructions to determine if the program instructions are
13 infected.

1 2. (Original) The method of claim 1, wherein the step of analyzing the
2 program instructions comprises an investigation of the contents of instructions within
3 code fragments.

1 3. (Original) The method of claim 1, wherein the step of analyzing the
2 program instructions comprises inserting decrypted program instructions into a virus
3 detection manager.

1 4. (Original) The method of claim 3, further comprising the step of:
2 releasing program instructions from the virus detection manager when infected
3 program instructions are not detected.

1 5. (Original) The method of claim 3, wherein the step of analyzing the
2 program instructions comprises performing a signature comparison with the contents
3 of the code fragments.

1 6. (Original) The method of claim 3, wherein the step of analyzing the
2 program instructions comprises monitoring the behavior of the contents of the code
3 fragments in a virtual computing device.

1 7. (Original) The method of claim 3, wherein the step of analyzing the
2 program instructions comprises applying a plurality of tests on the contents of the
3 code fragments in a virtual computing device.

1 8. (Original) The method of claim 4, further comprising the step of:
2 processing the released program instructions in computer hardware.

1 9. (Currently Amended) A system for detecting infected program
2 instructions in active software applications, comprising:
3 means for intercepting program instructions designated for execution within a
4 computing device;
5 means for transforming the program instructions;
6 means for determining when the intercepted program instructions have not
7 been processed by the computing device responsive to an association between native
8 application code from the active software applications and one or more analogues that
9 have been cached within a dynamic execution layer inserted between a processor and
10 program instructions; and
11 means for analyzing the intercepted program instructions that have not been
12 processed by the computing device prior to forwarding the intercepted program
13 instructions to computer hardware.

1 10. (Original) The system of claim 9, further comprising:
2 means for gaining control over execution of program instructions.

1 11. (Original) The system of claim 9, further comprising:
2 means for executing program instructions.

1 12. (Original) The system of claim 9, wherein the means for intercepting
2 comprises a dynamic execution layer interface (DELI).

1 13. (Original) The system of claim 9, wherein the means for analyzing the
2 intercepted program instructions comprises a virus detection manager.

1 14. (Original) The system of claim 13, wherein the virus detection
2 manager comprises a controller configured to apply a plurality of virus detection tests
3 over the contents of the intercepted program instructions.

1 15. (Currently Amended) A virus detection program stored on a computer-
2 readable medium, comprising:
3 logic configured to intercept program instructions;
4 logic configured to transform the program instructions;
5 logic configured to determine if the intercepted program instructions have not
6 been processed by a computing device responsive to an association between
7 application code and one or more analogues that have been cached within a dynamic
8 execution layer inserted between a processor and program instructions; and
9 logic configured to determine when the intercepted program instructions that
10 have not been processed by the computing device are infected with a virus.

1 16. (Original) The program of claim 15, further comprising:
2 logic configured to gain control over execution of intercepted program
3 instructions.

1 17. (Original) The program of claim 15, further comprising:
2 logic configured to execute program instructions.

1 18. (Original) The program of claim 15, further comprising:
2 logic configured to forward non-infected intercepted program instructions to
3 the computing device.

1 19. (Original) A computer system, comprising:
2 a processor;
3 an execution memory;
4 a dynamic execution layer interface (DELI) residing between at least one
5 application and the processor, wherein the DELI comprises:
6 a core configured to cache and execute certain application code
7 fragments;
8 an application programming interface configured to provide access to
9 caching and executing functions of the core to a virus detection manager; and
10 a system control and configuration layer configured to provide policies
11 for operation of the core.

1 20. (Original) The system of claim 19, wherein the virus detection
2 manager is configured to apply at least one virus detection test on the contents of
3 application code fragments.

1 21. (Original) The system of claim 19, wherein the core is configured to
2 process executable application code fragments from the at least one application that
3 have not been previously sent to the processor.

1 22. (Original) The system of claim 21, wherein the virus detection
2 manager controls whether application code fragments are released to the processor.

1 23. (Original) The system of claim 22, wherein application code fragments
2 that contain at least one virus signature are not released to the processor.

1 24. (Original) The system of claim 22, wherein application code fragments
2 that behave in a manner consistent with known virus attacks are not released to the
3 processor.